

AMENDMENTS TO THE CLAIMS

LISTING OF CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

1. (currently amended) ~~An In a dynamic programming system, the~~
improvement in a dynamic programming system, comprising:

a tree data structure implemented in a computer-readable memory accessible by a processor, said tree data structure having a plurality of nodes that topologically define a root node and a plurality of parent-child generations, including a deepest child generation that is topologically furthest from the root;

a traversal algorithm implemented by said processor, said algorithm traversing said nodes based on a set of traversal rules whereby nodes of a given generation are processed before the parent nodes of said given generation are processed, the deepest child generation is processed first, and traversal among nodes of each generation proceeds in the same topological direction, wherein said traversal algorithm includes a dynamic programming process that assigns probability scores to nodes that are traversed using a forward recursion, thereby computing the scores based only on knowledge of child nodes.; and

a mechanism for designating selected ones of said nodes as active nodes, wherein said active nodes have a probability score above a pre-determined search threshold, said probability score being determined from information sourced only from the child generation nodes, wherein said traversal algorithm only traverses said active nodes.

2. (original) The improvement of claim 1 wherein said tree data structure is encoded in said memory as a flat representation in which nodes of each generation occupy contiguous memory locations.

3. (original) The improvement of claim 1 wherein said tree data structure is encoded in said memory with parent-child generations being represented through linked list.

4. (original) The improvement of claim 1 wherein said tree data structure is encoded in said memory as a flat representation in which nodes of each generation occupy contiguous memory locations and have indicia designating the topological boundary between children of the same parent.

5. (cancelled)

6. (original) The improvement of claim ~~5~~ 1 wherein said mechanism for designating selected ones of said nodes as active nodes comprises an active envelope data structure associated with said tree data structure.

7. (original) The improvement of claim 1 further comprising a mechanism for designating selected ones of said nodes as active nodes and wherein said traversal algorithm includes a traversal rule whereby only active nodes are processed.

8. (original) The improvement of claim 1 wherein said tree data structure is a lexical tree representing a lexicon.

9. (original) The improvement of claim 1 wherein said tree data structure is a lexical tree representing the lexicon of a speech recognizer.

10. (currently amended) ~~An In a dynamic programming system, the~~ improvement in a dynamic programming system, comprising:

a tree data structure implemented in a computer-readable memory accessible by a processor, said tree data structure having plurality of nodes, including a root node, child generation nodes and parent generation nodes;

a mechanism for designating selected ones of said nodes as active nodes, said mechanism for designating selected ones of said nodes as active nodes defines an active envelope and uses a set of rules to propagate the active envelope, wherein said active nodes have a probability score above a pre-determined search threshold, said probability score being determined from information sourced only from the child generation nodes; and

a traversal algorithm implemented by said processor, said algorithm traversing said nodes based on a set of traversal rules whereby only said active nodes that are determined as having a probability score above the pre-determined search threshold are traversed, wherein said traversal algorithm includes a dynamic programming process that assigns probability scores to nodes that are traversed using a forward recursion, thereby computing the scores based only on knowledge of child nodes.

11. (original) The improvement of claim 10 wherein said mechanism for designating selected ones of said nodes as active nodes comprises an active envelope data structure associated with said tree data structure.

12. (original) The improvement of claim 10 wherein said traversal algorithm includes a dynamic programming process that assigns a likelihood score to nodes that are traversed.

13. (original) The improvement of claim 12 wherein said mechanism for designating selected ones of said nodes uses said likelihood score to designate said active nodes.

14. (original) The improvement of claim 10 wherein said traversal algorithm includes a dynamic programming process that assigns a likelihood score to nodes that are traversed and wherein nodes are designated as active nodes if their likelihood score is above a predetermined threshold.

15. (original) The improvement of claim 14 wherein said predetermined threshold is calculated based on the highest likelihood score.

16. (previously presented) The improvement of claim 10 wherein said mechanism for designating selected ones of said nodes as active nodes, and uses said set of rules to propagate the active envelope by removing nodes that have a likelihood score below a predetermined threshold.

17. (previously presented) The improvement of claim 10 wherein said a mechanism for designating selected ones of said nodes as active nodes, and uses said set of rules to propagate the active envelope by inserting nodes that have a likelihood score above a predetermined threshold.

18. (original) The improvement of claim 17 wherein said set of rules for inserting nodes guarantees that the nodes in said active envelope are sorted by their topological index.

19. (previously presented) The improvement of claim 10 wherein said processor employs a cache and wherein said tree data structure is encoded in said memory such that traversal of said tree proceeds into said cache.

20. (cancelled)